Design of a Hyper Media Tool to support Requirements Elicitation Meetings

Haruhiko Kaiya

School of
Information Science

Japan Advanced Institute
of Science and Technology,
Hokuriku

Ishikawa, 923-12 Japan
kaiya@jaist.ac.jp

Motoshi Saeki

Graduate School of Information
Science and Engineering

School of Information Science

Tokyo Institute of Technology

Tokyo, 152 Japan
saeki@cs.titech.ac.jp

Koichiro Ochimizu

School of
Information Science

Japan Advanced Institute
of Science and Technology,
Hokuriku

Ishikawa, 923-12 Japan
ochimizu@jaist.ac.jp

Abstract

In this paper, we introduce a hyper media tool for requirements elicitation meetings. We consider requirements elicitation meetings as a consensus making process among participants who have their own roles. Participants in the meeting usually repeat the following activities to finish the final specification;

- *Preparing agenda and/or final specification for the next meeting while referring both his memory and secretary's minutes if accessible.*
- *Pursuing the arguments in a meeting while referring either the agenda for the meeting or minutes and his memory about the previous meetings.*

From the observation of several real meetings, most of the final specifications were inconsistent, and the unnecessary and redundant communication had occurred in the meetings because a large amount of verbal data in the meetings makes each participant's memory and meetings' minutes, incomplete and ambiguous. From this point of view, our tool gives the participants the following three facilities to reduce them;

1. *The plain record of the meetings.*
2. *The repository for minutes and agenda extracted from the record.*
3. *The multi-modal and graphical user interfaces for referring the repository.*

Such facilities can avail for the participants to develop both suitable minutes and agenda. Our tool will improve the efficiency of a consensus making process by suppressing the unnecessary and redundant communication.

1 Introduction

In the upper stream of software processes, verbal communication plays an important role because many workers have verbal interaction to perform their activities effectively. In the Gary's paper [1], he reported that "the participants in 8 of these 10 meetings spent on average more than 90% of the time talking to each other". In this paper, we introduce a CASE tool for workers who participate in the face-to-face meetings to develop software requirements specification. Activities in the meetings has the following features;

- It is difficult to manage the information in the meetings because the participants use informal and non-structured form of information.
- It is difficult to formalize the information in the meetings because methods and/or languages for formalizing the information will hinder the participants' activities such as thinking processes and verbal communications.
- It is difficult to record and to refer the informal and non-structured form of information. Such difficulty causes inconsistent products and the unnecessary and redundant communication.

To make participants overcome such difficulties, our tool has the following features;

- Our tool does not hinder the users' activities such as thinking processes and verbal communications. Several tools often force complex and frequent manipulations on the users and hinder the users' activities
- Workers can refer the necessary information from the informal and non-structured record both easily and quickly any time he wants, with the use of multi-modal user interfaces of our tool.

- Our tool is independent to the specific working manners of design but depend on the characteristics of the face-to-face meetings. Hence, users can follow familiar working manners with our tool. Some tools often force only some specific manners, e.g. design methods [2],[3] and formal languages [4], [5], on the workers. But such manners can not always support their activities effectively because workers are not always familiar with such manners.

There are various types of group activities, for example, meetings and multi-author documentation[6], and several tools for supporting the activities have been studied and developed. These tools are usually based on the general and theoretical characteristics of cooperation, e.g. speech act theory[7], IBIS[8] and QOC[9]. As Curtis pointed out[10], these tools could not improve performance of group activities because they are based on the general and theoretical characteristics. In our research, we have clarified the characteristics of workers' cooperation by observing the activities in the actual requirements elicitation meetings. From the implication of these observations, our tool was designed.

This paper is organized as follows. In Section 2, we clarify the activities for the participants, which our tool can support during/after meetings.

In Section 3, we introduce supporting facilities of our tool. Our tool provides three major facilities for the participants;

- The recorder in the meetings plainly: Our tool can record the utterances and gestures of the participants through the meetings semi-automatically in the computer systems.

- The repository for minutes and agenda extracted from the record: Our tool can structure the multi-modal and hyper text form of information from the record of the meetings to prepare the minutes of the meetings and the agenda for the next meetings.

- The graphical and multi-modal user interfaces for referring the repository: The tool can help users confirm the contents of the discussions, grasp the stream of discussion, and check the agenda at any time, during/after the meeting.

In Section 4, we present an experiment of our prototype system for evaluating our tool. Finally, we summarize our tool and discuss usage of our tool in the distributed working environment.

2 Which activities can our tool support?

In the requirements elicitation stage, workers participate in the face-to-face meetings with their own role, e.g. customer, user, designer, chairman and secretary. And they mainly communicate the informations to the other participants by the utterances and the gestures. Informations consist of many kinds of topics about the target system and about the development processes. A customer should communicate, for example, the limit of the financial and time resources, users may communicate his works which would be supported by the target system, designers can communicate the technical advice about the hardware, and chairman can communicate the time limit in a meeting. Through the communications, participants normally develop the following intermediate artifacts for completing the requirements specification document.

- Agenda: the list of prepared informations for the target system and for the development processes. Normally, several participants prepare their agenda respectively for a meeting.

- Minutes: the record of meetings which are already finished. Normally, mutual agreements are mainly listed in the minutes.

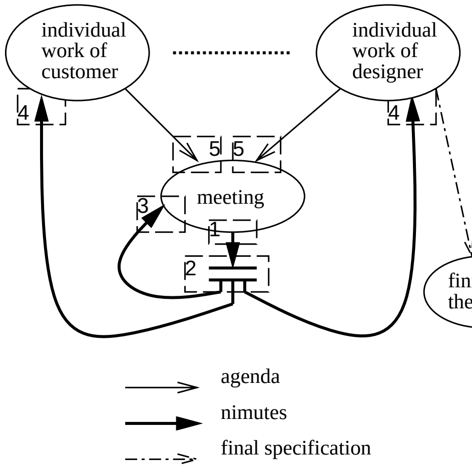


Figure 1: Participants' activities represented by Data Flow Diagram

From the view of cooperation, we regard the participants' activities as a simple data flow diagram shown in Figure 1. We consider that computer tools can support the five kinds of activities shown by the dashed rectangle in Figure 1;

1. Recording the contents of meeting: Computer tool can record the contents of meetings directly and automatically with the multi-modal user interfaces, e.g. sound, video. Such recorded data are plain but not good for quick access.

2. Storing and Structuring the records: Computer tool can store the raw recorded data in his storage unit. And it can structure the information which is useful for accessing the record quickly, for example hypertext form.

3. Providing user interfaces to refer the records in the meetings: Computer tool can avoid the the

unnecessary and redundant communication in behalf of the secretary

4. Providing user interfaces to refer the records in the individual work: Computer tool can avoid inconsistency in the products such as agenda in behalf of the secretary.

5. Providing user interfaces to refer the agenda in the meetings: Computer tool can check whether topics in the prepared agenda are discussed or not in behalf of the chairman.

From the observation of several real meetings[11], the final specification had been inconsistent, and the unnecessary and redundant communication had occurred in the meetings because of vague memory and minutes[11]. So these computer supports are necessary for the consistent products and efficient communication.

3 What kind of facilities should our tool provide?

In this section, first we present the working environment and hardware equipments of our tool. Participants in the meetings can use our tool not only in the meeting but also for the individual work. Second, we introduce the data structure of the meeting records along with the agenda. Third, we explain how tool users structure the useful information for quick access from the recorded data. We call such structured data as *hyper minutes* because the data are stored with the hyper text form. And in the last two subsections, we introduce the graphical and multi-modal user interfaces for retrieving the hyper minutes both as minutes and as agenda.

3.1 Working environment of our tool

As mentioned in Section 2, participants in the meetings works not only in the meetings but also before/after the meetings to accomplish meetings effectively. So our tool should not only support the activities in the meetings but also those before/after the meetings, where participants would summarize the minutes of the previous meeting, and where that would describe the agenda for the next meeting. We define the two working modes for using our tools, *synchronous mode* and *asynchronous mode*, in Figure 2; the former for the activities in the meetings, the latter for those before/after the meetings.

In synchronous mode, participants hold the meeting, discuss the contents of agenda, and decide the requirements about the target system. In asynchronous mode, several participants summarize the contents of discussions in the previous meeting to make minutes or final documents and to prepare the agenda for the next meeting respectively. To repeat each mode of works in turn, participants can refine stepwise the requirements specification.

Figure 3 shows a simple example of working processes with our tool. In this example, a requirements document was developed by a customer, a designer and a secretary. First, the customer prepared his needs as an agenda(white rectangle 1), and meeting1(gray rectangle 1) was held. From the recorded histories of meeting1, the secretary summarizes them and describes minutes for referring the other participants (white rectangle 2). The customer rewrote his agenda by taking account of the minutes(white rectangle 3), and the designer described the first version of requirements document as an agenda respectively(white rectangle 4). And meeting2(gray rectangle 2) was held for negotiating the requirements, the secretary described minutes again(white rectangle 5), the designer described final version of requirements(white rectangle 6) and finally the customer approved the requirements(while rectangle 7).

In each mode, especially in the synchronous mode, manuplation on the tool should not hinder the participants' activities. In the meeting, participants normally pursue the arguments with the verbal communication, gestures, and descriptions on the blackboad or on the papers if necessary. Tools can *support* the participants to store such informations as textual or structured-graphical form in the computer system, but such *supports* would interrupt the argumentation. To overcome this obstacle, we equip our tool with multimodal user interfaces for managing the sound and graphical information directly. In Figure 2, each participant in synchronous and asynchronous mode, has their own microphone and headphone for the verbal information. By these equipments, they can be recorded by his own utterances in the computer system, and can refer utterances already recorded. Figure 4 shows the user interfaces in detail. And in Figure 2, gestures, and descriptions on the blackboad or on the papers can be recorded by the video camera which is connected to the computer system. Our prototype system can not recognize the words automatically, but the break between the utterances semi-automatically.

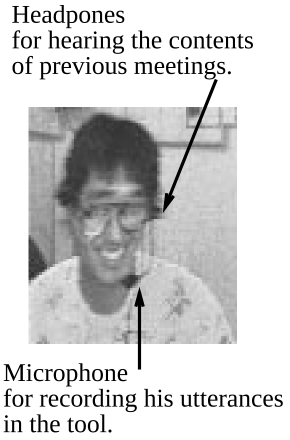


Figure 4: User interfaces for utterances in the meetings

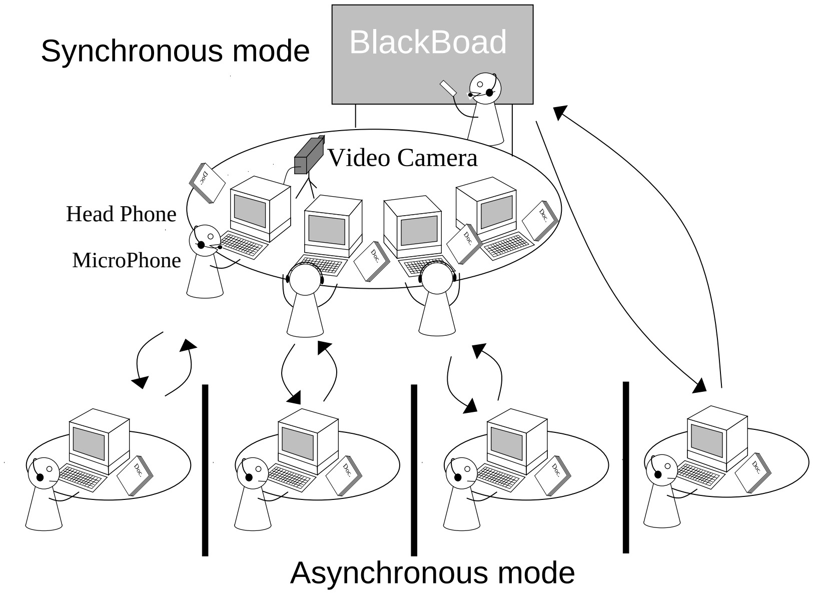


Figure 2: Two major working modes: Synchronous and Asynchronous mode

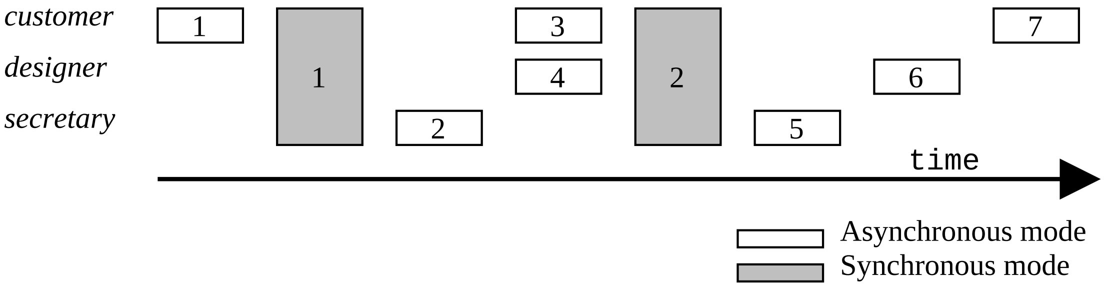


Figure 3: An example of working processes with our tool

3.2 Data structure for storing meeting records

Participants in the meetings want to refer the records of the meetings both correctly and quickly, for pursuing the arguments efficiently in a meeting, and for constructing minutes and agenda correctly before/after the meeting. To satisfy such requirements, our tool has the data structure of the records as follows;

- Raw recorded data, such as the utterances and the gestures, are represented in the structure. The recorded data has the hierarchical and temporal structure. This structure helps participants access the meeting record quickly.
- The artifacts, such as textual form of the minutes, the agenda and/or the final specification, are represented.
- The participants' responsibilities and/or authorships for both the products and the recorded data, are represented.
- The relationships between the raw data and the artifacts are represented for accessing the design rationale from the recorded data plainly.

The structure is shown in Figure 5. This figure is represented in Schema Definition Set format of PCTE (Portable Common Tool Environment) Object Base [13]. This structure consists of major three parts shown by the gray rectangle in Figure 5 as follows;

1. Project part: "Action" type in Figure 5 denotes recorded data, i.e. voice, images, and so on. We can recognize the actor of an action from the link between "worker" and "action", which any participants create in the meeting. The instances of

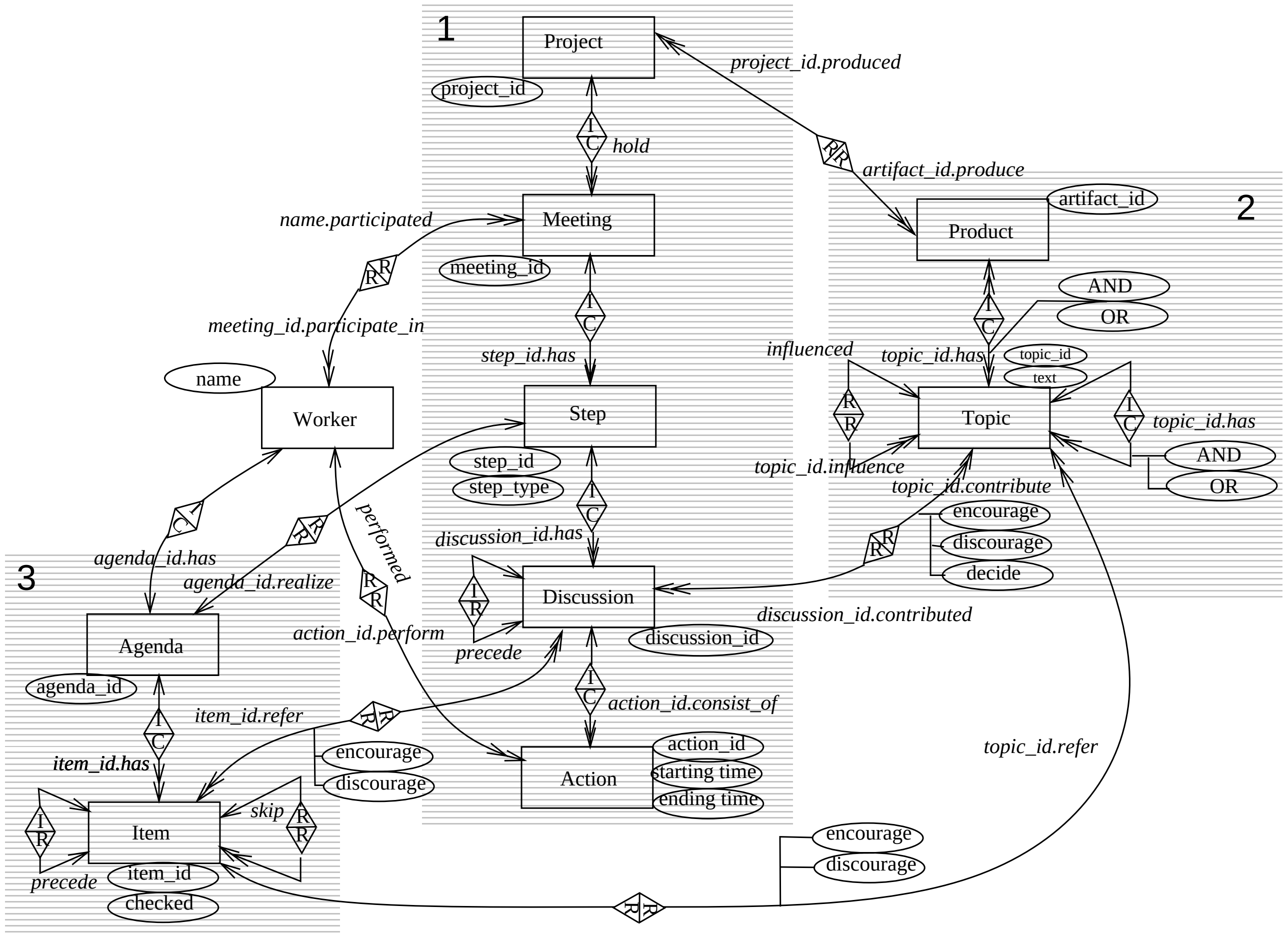


Figure 5: Data Model for Structuring Meeting Records and Products

actions are grouped into a discussion. Concretely a discussion consists of a proposal, a question, an answer, an agreement and so on about a topic. The instances of discussion are semantically and temporally connected to each other. The link “precede” between “discussion” type denotes this temporal relationship. That is to say, the records in the meeting can be modeled as a sequence of discussions. The steps consists of discussions and this relationship can be defined by the link “*discussion_id.has*”. The attribute “step_type” of the step has the value such as “customer’s explanation”, “review and discussion”, and “schedule development”. Normally, activities in each step follow the topics in the agenda.

2. Product part: This part represents final specification and/or minutes of the meetings. We assume such kinds of products as the indented text with figures. A “topic” is constituent of the products, e.g. a sentence, a figure and a table. And the topics are hierarchically composed of sub-topics in AND-OR tree form.

   The relationship “influence” (influence and influenced links) between topics is used for representing their semantical dependency. Assume that a conclusion of a topic is changed in a meeting, this change should propagate to the other topics. We can know which topics might be effected by the change by means of tracking the influence relationship. In our prototype system[11], this relationships are automatically created according to the temporal order of the occurrence of the topic in the meetings, that is to say, if topic `B` is discussed after the discussion about topic `A`, our tool creates the influence a relationship between the `A` and `B` and a relationship between the `B` and `A`.

   A discussion may encourage (support) or discourage (object to) topics. Otherwise it may declare a topic as a conclusion which is officially decided. The attributes “encourage”, “discourage” and “decide” of the links from the discussion to the topics express the semantical roles of the topics in the meetings. For example, assume that the last discussion in the meeting "decides" the topic. It will be a final conclusion in the project if no discussions "discourage" it in the further meetings.

3. Agenda part: Each participant can prepare the "agenda" for the next meeting. Agenda consists of the sequence of "item", which represents the scenario in the next meetings. Each item corresponds to the each participants' idea, opinion or material for encouraging the discussions in the next meetings, and has the attribute "checked" for checking if it would be discussed.

   Participants can pursue the arguments in a step according to the "agenda". An item may encourage or discourage topics in the same way of a discussion, but it can not declare a topic as a conclusion because it is nothing but a personal decision. Each item may be encouraged or discouraged through the discussion, and it can produce some parts of conclusion, topics. The relationship "skipping" between the items is used for skipping items, which precondition item is discouraged.

3.3 How to construct the hyper minutes and agenda

For the sake of useful access to the meeting record, users of our tool should structure the information from the recorded data along the data structure in Figure 5. We introduce the standard procedure for structuring. Our tool provides editing facilities for this procedure.

1. Identify steps: As mentioned in subsection 3.2, participants pursue the arguments along an agenda in a step. And each agenda would be used in turn if more than one participant is prepared his own agenda. Users of our tool can easily identify the steps to keep track of the major actor in the meetings. *Action editor* in Figure 6 support the user to keep track of the major actor in the meetings.

   In the action editor, the names of participants are displayed in the leftmost column of the action editor window. In this case, we have three participants, "gotoh", "saeki", and "miura", who participated in the meeting#1. Many horizontal bars appear in the action editor window, and they denote the actions of the participants such as utterances or gestures in the meeting. You can display or listen to an action when you push the corresponding bar.

   In this example, participant named "saeki" is the major actor hence his agenda would be used in this period of time. Users of our tool may make relationship between his agenda and this period of time, which denotes a step.

2. Identify discussions: A discussion is a sequence of actions, where participants mentioned a specific part of the products. Concretely it consists of a proposal, a question, an answer, an agreement and so on about a topic. Users can group the actions into discussions on the action editor and select both the start and the end point of discussion by mouse clicks while displaying or listening to an action when you push the corresponding bar. A shaded box in Figure 6 is an example of the created discussion. All actions included in the box constitute the discussion. The window also tells that the starting and ending time of the discussion are 14:55:41 and 14:56:42 in the right bottom.

3. Create topics: Users can create a topic as a textual form while selecting a discussion. Normally, key word or sentence in the discussion is used for the text in the topic. Before creating the text, users should select whether the discussion encourage, discourage or decide the topic. Note that users can create topics without the relationship to a discussion.

4. Make relationship among the topics: Users can make has-a relationship among the topics freely, but the ordering of the discussions would suggest the relationship. For example, subpart of a topic would be discussed just after the topic.

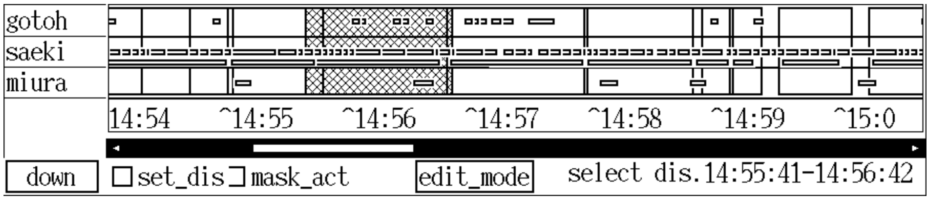
According to our experience, it is hard to complete this procedure only in the synchronous mode. So, this procedure is mainly accomplished in the asynchronous mode. In an experiment, it took about two hours for a worker to structure a information from a 30 minutes meeting. It seems too long but we think it is necessary to construct the minutes and the agenda which are fully reflected the contents of the discussions. We are now exploring some methodologies to construct artifacts from the meeting records efficiently.

For the sake of efficient progress in the meetings, participants should prepare the agenda which is related to the meeting record. We introduce the standard procedure for preparing the agenda. Our tool also provides editing facilities for this procedure.

1. List the items: A user may list the items which he want to discuss in the next meeting. He may prepare the evidence which supports his opinion in the item.

2. Create skip-relationships among the items: A user may create "skip" relationships between the premise and the conclusions. If an item which is premise is rejected in the meetings, he can easily skip over the items which are conclusions with this relationships.

3. Make relationship between an item and discussions: If a user want to mention an existing discussion in an item, he should make relationship between the discussion and the item. And he should select whether the item encourage or discourage the discussion. For this relationship, The user can easily refer the discussions, which he want to mention in his agenda. Note that the user can create item without this relationship.



select dis.14:55:41-14:56:42

Figure 6: Action editor

4. Make relationship between an item and topics: The same as above.

Normally, this procedure is accomplished in the asynchronous mode.

3.4 How to use the hyper-minutes as a minutes

Because our tool supports the participants to access the informations about the meetings which have been already finished, inconsistent products and the unnecessary and redundant communication can be suppressed. Concretely, our tool provides viewer for the instance of "project" and "product" parts in Figure 5.

Figure 7 shows an example of a screen of our tool. As shown in the middle right pop-up menu whose label is "mtool", we have four commands at the top level of the tool. The "viewer" command in the menu initiates a *browser* for the hierarchy structures of the topics. The tree structure of the topics is shown in the left window in Figure 7. It shows the meeting for discussing about *"the supporting tool for a program chairperson of the international academic conference such as CASE'95"*. When you select "Editor" in the menu, the tool initiates an *action editor* mentioned in Subsection 3.3, for referring actions, discussions and participants in the meeting.

Consider how to manipulate the relationships between topics and discussions on the browser and the action editor. If you select Topic30 on the browser, the topic window appears as shown in the bottom right part of Figure 7. You click on the "contributed" button by your mouse. You can see a discussion window where the discussions related to the Topic30 are displayed. In Figure 7, the window tells us that a discussion is related to the topic30 through the "contributed" link whose attribute value is "encourage". That is to say, the discussion from 14:55:41 to 14:56:42 encourages the topic about "sending papers and reviewer forms to the reviewers". The corresponding box on the action editor window is shaded. The navigation to discussions or to topics is based on the SDS shown in Figure 5.

3.5 How to use the hyper-minutes as a agenda

The relationships between the product and project in the Figure 5 only represent the activities about the meetings which were already finished. In contrast, the relationships between the agenda and the project in the Figure 5, can represent the activities which are planed by the each participant. Our tool provides the following information through the graphical and multi-modal user interfaces;

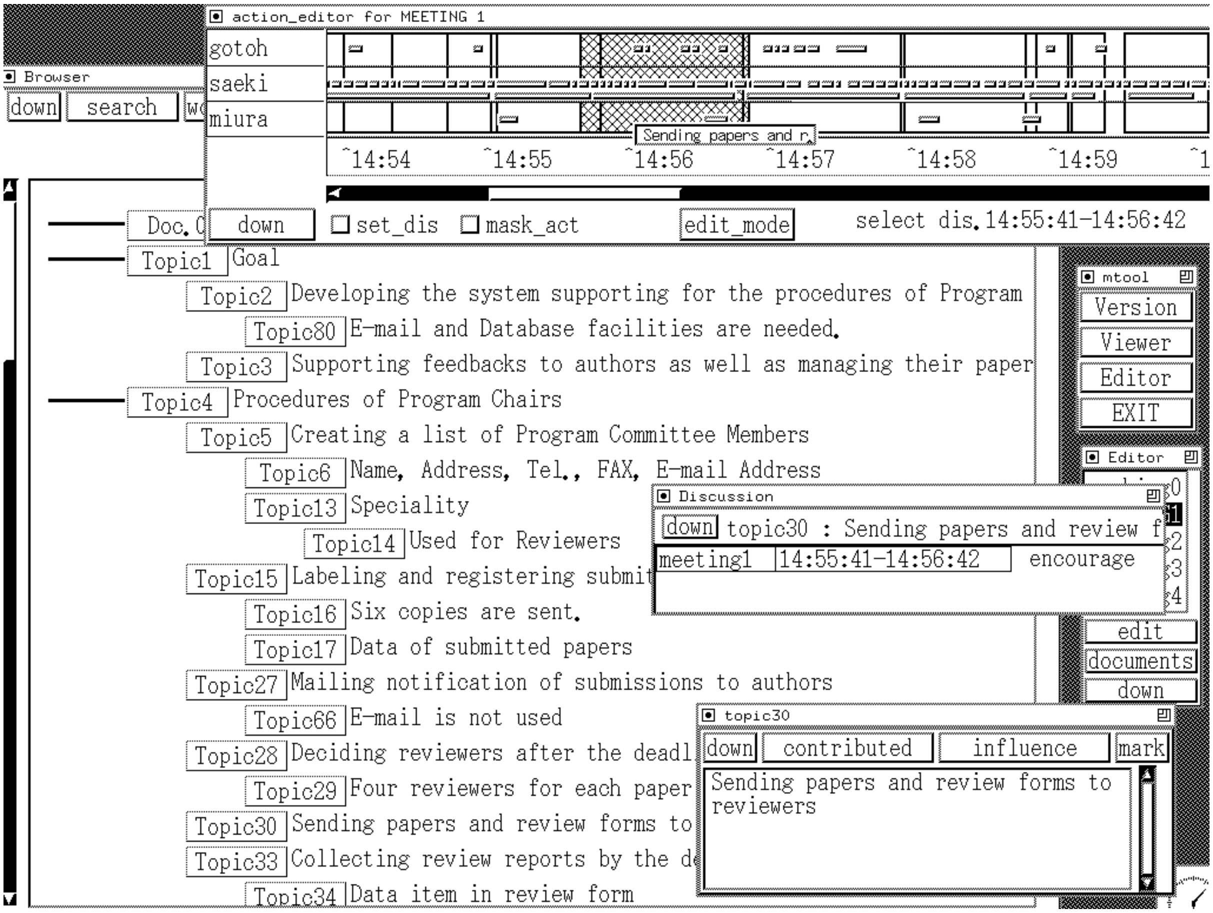
- The discussions which is mentioned in each item in the agenda.

- The flags where an item in the agenda is already discussed or not.

The editor for preparing the agenda is shown in the top-left in Figure 8. Each participant can describe agenda as the sequence of his idea, opinion and/or resources for the next meeting. The constituent of the sequence is named "item" in Figure 5. In Figure 8, we can see five items in the agenda, from #5 to #9. Workers can make relationship between his item and topics in the artifact, which are already exists. Using this relationships, each participant can think over the topics, and collect the grounds and/or the evidences for persuading his opinion. In Figure 8, item 5, "FTP service is preferred", discourages the Topic80, "E-mail and Database facilities are needed", and its reason is listed in the following item, item 7. And using item 9, this participant proposes to the PC tool with its look and feel (left down in Figure 8, of course, this picture is nothing but an example of tool's look and feel).

In the meeting, participants can discuss the items in the agenda successively, and easily identify the step in Figure 5. Normally, more than one participant would prepare his own agenda for the next meeting. And each agenda would include an item which is semantically similar to one in the other participant's agenda. Participants can follow several agendas simultaneously with our tool, and more the one items in the different agenda, can be discussed together.

Figure 9 shows an example of instance of Figure 5 with two parallel steps. In this example, participants follow the worker1's agenda, "agenda1", from the discussion#21, "dis.21", in "step1". "Dis.22", "dis.25" and "dis.27" in "step1" are not prepared by the owner of "agenda1", "worker1" in advance, because they are not related to the items in "agenda1". "Dis.22" and "dis.27" would appear in "step1" because of the context in the meeting, e.g. suggestion of a participant. In "dis.23", "worker2" find a topic which is similar to the topic discussed in "dis.23", in "item c". So participants can follow worker2's agenda simultaneously. Because each item has the attribute named "checked"(see Figure 5), "worker1" may not forget



select dis.14:55:41-14:56:42

Doc.C

Topic1 Goal
Topic2 Developing the system supporting for the procedures of Program
Topic80 E-mail and Database facilities are needed.
Topic3 Supporting feedbacks to authors as well as managing their paper
Topic4 Procedures of Program Chairs
Topic5 Creating a list of Program Committee Members
Topic6 Name, Address, Tel., FAX, E-mail Address
Topic13 Speciality
Topic14 Used for Reviewers
Topic15 Labeling and registering submit
Topic16 Six copies are sent.
Topic17 Data of submitted papers
Topic27 Mailing notification of submissions to authors
Topic66 E-mail is not used
Topic28 Deciding reviewers after the deadl
Topic29 Four reviewers for each paper
Topic30 Sending papers and review forms to
Topic33 Collecting review reports by the d
Topic34 Data item in review form

topic30 : Sending papers and review f

meeting1	14:55:41-14:56:42	encourage

Sending papers and review forms to reviewers

Figure 7: Example of the Supporting Tool with secretary function

"item a" and "item b". In Figure 9, the "checked" attribute is represented by a white or black circle left under the "item" rectangle; the former for the not checked item, the latter for the checked item. "Dis.25", which is not prepared in "agenda1", would appear in "step1" because "step1" would be semantically synchronized with "step2". Of course, we can construct the steps in Figure 5 without agenda. Currently, our prototype system supports only this way.

4 Evaluation

In this section, we present an experiment of our tool for evaluating the design of our tool. As mentioned in Subsection 3.5, Our prototype system do not support agenda part completely.

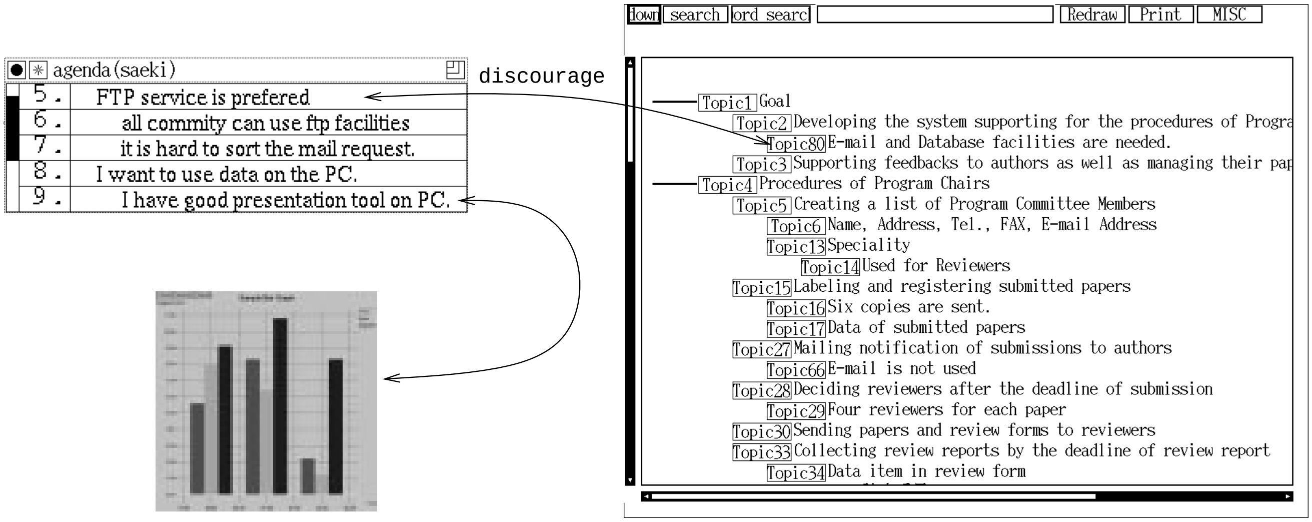
In this experiment, two customers and a designer held three meetings for developing the requirements specification. The target system is *"the supporting tool for a program chairperson of the international academic conference such as CASE'95"*. Participants spent about an hour in each meeting.

From the observation of this experiment, we have find the following features of our tool;

- The access for the meeting record in the meeting: It took only about 2% of all meeting time for participants to listen the record of the previous meetings. But workers could dissolve the vagueness of the topic easily.

- The access for the meeting record while describing the agenda: It took about 35% of all working time for participants to describe agenda after the meetings. If they could not use our tool, they should fill up the rest of the time for his own memory and memos about the previous meetings, hence the final specification would be inconsistent, and the unnecessary and redundant communication would have occurred in the meetings.

- The access for the meeting record while describing the minutes: As mentioned in Subsection 3.3, it took about two hours for a worker to structure the information from a 30 minutes meeting. If participants did not use this tool, it would take fifteen hours. We estimate this cost of time from the cost of analysis for observing actual soft-



5. FTP service is prefered
6. all commity can use ftp facilities
7. it is hard to sort the mail request.
8. I want to use data on the PC.
9. I have good presentation tool on PC.

```
Topic1 Goal
    Topic2 Developing the system supporting for the procedures of Progra
        Topic80 E-mail and Database facilities are needed.
    Topic3 Supporting feedbacks to authors as well as managing their pap
Topic4 Procedures of Program Chairs
    Topic5 Creating a list of Program Committee Members
        Topic6 Name, Address, Tel., FAX, E-mail Address
        Topic13 Speciality
            Topic14 Used for Reviewers
    Topic15 Labeling and registering submitted papers
        Topic16 Six copies are sent.
        Topic17 Data of submitted papers
    Topic27 Mailing notification of submissions to authors
        Topic66 E-mail is not used
    Topic28 Deciding reviewers after the deadline of submission
        Topic29 Four reviewers for each paper
    Topic30 Sending papers and review forms to reviewers
    Topic33 Collecting review reports by the deadline of review report
        Topic34 Data item in review form
```

Figure 8: Example of agenda

ware processes[11]. In each observed process, a transcription[12] of the discussion was described for constructing the product which was fully reflected the contents of the discussion. From this result, our tool contributes to the efficiency for developing correct and useful minutes.

- The influence of a modification: In this experiment, only one topic modified in the meeting. Several topics were listed by our tool automatically, and participants modified all of the topics for avoiding inconsistency.

- The necessity for multiple agendas: We find that most of participants described his own agenda and that they wished to advance the meeting along his own agenda, but our prototype system could use only one agenda as an input for a meeting.

  Investigating the two customers' agendas, about half items were similar to items in designer's agenda. From this result, if the customers and the designer could be share the similar items in each agenda, it would be took about two third hours for participants to spend the time in the meetings.

5 Conclusion

In this paper, we introduced a hyper media tool to support meetings, where requirements specifications are produced. From the observation of the ordinary meetings, we designed

the recorder for actions in the meetings plainly,

the repository for structuring the minutes and agenda frm the the plain record,

and the multi-modal and graphical user interfaces for referring the minutes and agenda.

We believe requirements elicitation can be accomplished mainly in the face-to-face form, such as meetings. So our tool can support activities in the requirements elicitation because workers can overcome the difficulties in the meetings with our tool.

Recently, we have to perform cooperative activities in the distributed working environment. Participants in such environment will suffer several disadvantages such as disadvantage for sharing the products, their working time, working space and so on.

So we are now going to enlarge the capability of our tool to support cooperative activities in the environment. Using the groupwares for covering the disadvantages, e.g. MERMAID[14], TEAM WORKSTATION[15] and IP multicast technology[16, 17], our tool can be also helpful in such environment. In practice, we have used our tool with WB, NV and VAT [16, 17], in our distributed environment, where one area is 300km away from another area, for confirming the adaptability of our tool.

Acknowledgements

The authors would like to thank the members of *User Oriented Software Process Group*, International Institute for Advanced Study of Social Information Science, FUJITSU LABORATORIES LTD, for discussions and insightful comments which led to some of the issues raised in this paper.

References

[1] Gary M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storrosten. Small group design meetings: An analysis of collaboration. *Human-Compter Interaction*, 7(3):pp.347–374, 1992.

[2] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen.

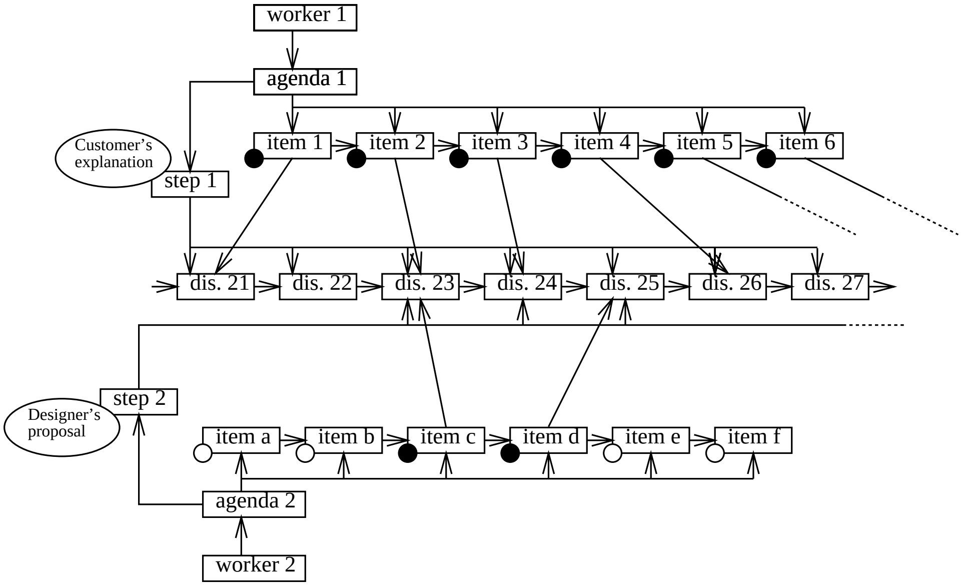


Figure 9: Instance of Figure 5 with two parallel steps

Object-Oriented Modeling and Design. Prentice-Hall, 1991.

[3] Grady Booch. Object-oriented development. *IEEE transactions on software engineering*, SE-12(2):211–221, Feb. 1986.

[4] Ben Potter, Jane Sinclair, and David Till. *An Introduction to Formal Specification and Z.* Prentice Hall International, 1991.

[5] C.B. Jones. *Systematic Software Development Using VDM.* Prentice Hall, 1986.

[6] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware : Some Issues and Experiences. *CACM*, 34(1), 1991.

[7] Terry Winograd. Where the action is. *BYTE*, 13(13), Dec. 1988.

[8] G. L. Rein and C. A. Ellis. rIBIS: a real-time group hypertext system. *International Journal of Man-Machine Studies*, 34(3):349–368, Feb. 1991.

[9] MacLean,A., Young,R.M., Bellotti,V.M.E., and Moran,T.P. Questions,options,and criteria: Elements of design space analysis. *HUMAN-COMPUTER INTERACTION*, 6(3 & 4):201–250, 1991.

[10] Bill Curtis. Implication from empirical studies of the software design process. In *Proceedings of an International Conference organized by the IPSJ to communicate the 30th Anniversary*, 1990.

[11] Haruhiko Kaiya and Motoshi Saeki. A Groupware for Face-to-face Meetings to Develop Software Specifications. In *InfoScience '93, International Conference organized by Korea Information Science Society*, pages 691–698. Korea Information Science Society, Oct. 1993.

[12] Charles Goodwin and John Heritage. Conversation analysis. *Annual Reviews Anthropoly*, Vol. 19, pp. 283–307, 1990.

[13] Lois Wakeman and Jonathan Jowett. *PCTE The Standard for Open Repositories.* Prentice Hall, 1993.

[14] Kazuo Watanabe, Shiro Sakata Kazutoshi Maeno, Hideyuki Fukuoka, and Toyoko Ohmori. Distributed Multiparty Desktop Conferencing System: MERMAID. In *CSCW90*, pages 27–38, Oct. 1990.

[15] Hiroshi Ishii. TeamWorkstation: Towards a Seamless Shared Workspace. In *CSCW90*, pages 13–26, Oct. 1990.

[16] Hans Eriksson. MBONE: The Multicast Backbone. *CACM*, 37(8):54–60, Aug. 1994.

[17] Michael R. Macedonia and Donald P. Brutzman. MBone Provides Audio and Video Across the Internet. *IEEE Computer*, 27(4):30–36, Apr. 1994.